REMARKS/ARGUMENTS

1.) Claim Rejections – 35 U.S.C. §101

The Examiner has maintained the rejection of claims 39-53 on the asserted basis that they are directed to non-statutory subject matter. The Applicants, again, traverse the rejection.

Claims 39-53 are drafted in "means-plus-function" format. The "means-plus-function" format is explicitly authorized by statute. Specifically, 35 U.S.C. §112, sixth paragraph, provides that:

> An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof. (emphasis added)

Thus, claims to a combination of elements that are expressed in a "means-plus-function" format are to be "construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof."

In response to Applicants' prior arguments, the Examiner merely responds that "[t]he 'means for' found in claims 39-53 are directed to software, which is not patentable subject matter." **The Examiner cites absolutely no support for the contention that software is not patentable**. The Applicants' claims are directed to methods and apparatuses that may be implemented, in part, by "software," but the claims are not directed to software *per se*. It is recommended that the Examiner review the new Interim Guidelines Regarding Patentable Subject Matter issued on August 24, 2009.

2.) Claim Rejections – 35 U.S.C. §102(a)

The Examiner has maintained the rejection of claims 27-53 as being anticipated by RFC 3321. The Applicants, again, traverse the rejections.

It is important to remember that anticipation requires that the disclosure of a single piece of prior art reveals **every** element, or limitation, of a claimed invention. Furthermore, the limitation that must be met by an anticipatory reference are those set forth in each statement of function in a claims limitation, and such a limitation cannot be met by an element in a reference that performs a different function, even though it may be part of a device embodying the same general overall concept. RFC 3321 fails to disclose each and every limitation of claims 27-53 and, therefore, those claims are not anticipated thereby.

In responding to Applicants' extensive arguments submitted in response to the prior office action, the Examiner vaguely asserts that a "shared state" disclosed by RFC 3321 is equivalent to the "state" recited in claim 27. The Examiner, however, does not address in any manner the additional claim elements relating to the claimed state as described in Applicants' arguments. The Applicants submit the following additional arguments to distinguish the claimed invention over the teachings of RFC 3321

RFC 3321 describes how to implement certain mechanisms in Signaling Compression ("SigComp"), RFC 3320, which can significantly improve the compression efficiency compared to using simple per-message compression. SigComp uses a Universal Decompressor Virtual Machine (UDVM) for decompression, and the mechanisms described in RFC 3321 are possible to implement using the UDVM instructions defined in RFC 3320. RFC3321 refers to extended operations of SigComp and, in particular, to specific types of compression: dynamic compression and shared compression (section 1). Generally, dynamic compression is compression relative messages sent prior to a current compressed message, whereas shared compression is compression relative messages received prior to a current compressed message (section 2, paragraphs 4 and 6 on page 3).

Starting with dynamic compression, when the compressor in the first endpoint compresses a message (m1), it uses information in a stored SigComp state (s0) (Fig. 2 and section 4.1). A new state (s1) is then generated based on the message (m1) and the previous state (s0). The compressed message (m1) is then forwarded to the second endpoint, where a corresponding state generation is performed using the received message (m1) and the state copy (s0) of the second endpoint (Fig. 2). Thus, in this dynamic compression, the state information is updated based on new messages. For this compression type to be implemented, however, both endpoints must first have access to an initial state (s0) based on which new states (s1, s2, s3) can be generated. RFC 3321 is **silent** about how this initial **state** will be **exchanged** between the endpoints in an efficient manner, enabling the endpoints to determine that the correct state has been successfully exchanged.

In shared expression, the so-called shared state is simply an uncompressed application message generated by one of the endpoints (section 2). A first endpoint saves the uncompressed version of the message (provided from its associated application) in a compartment dedicated to a second endpoint in its state memory (section 5.2). The message is then compressed and communicated to the second endpoint. This second endpoint calculates the shared state ID for this state (*i.e.*, for the received message). The calculated shared state ID is forwarded to the state handler in the second endpoint using the returned SigComp parameters (section 5.2, step (3)). The state handler compares this shared state ID (ID1) with a value (ID2) it has calculated for the current received and decompressed application message (section 5.2, step (4)). Thus, the **second endpoint determines both** the **ID1 and ID2**. If the identifiers match, the second endpoint will use this shared state (uncompressed received message) for compressing the **next** application message sent to the first endpoint (section 5.2, step (4)). This **shared state**, however, will **not** be saved in the second endpoint. Instead, it is forwarded up to the application in the second endpoint once it has been used in the single message compression. Thus, the received shared state is only used in the second endpoint for compression of the **single immediately following** message to be transmitted to the first endpoint.

There are, thus, several important differences between the Applicants' claimed invention and shared compression and states described in RFC 3321. Firstly, a shared state is an uncompressed application message and is only used for compressing **a single** following application **message** (and of course for decompressing this following application message in the other endpoint). Furthermore, although the shared state is stored in the first endpoint that created it (for the purpose of being used when decompressing the next compressed message to be received from the second endpoint), RFC 3321 does **not disclose** that the shared state is also **stored in the second endpoint**. On the contrary, once it has been used in a message compression in the second endpoint, the shared state is provided to the application in the second endpoint for further processing, as any other application message (remembering that the shared state is an uncompressed application message). As a consequence, a shared state is only applicable (once) in **one-way** message communication. This is described in the present application on page 26, lines 5 to 20.

As is clearly stated in RFC 3321 (section 5.2, steps (3) and (4)), the **second endpoint calculates** both the shared state ID, **ID1**, and the value, **ID2**, for the current received and decompressed message (which is identical to the shared state). As a consequence, the second endpoint will basically determine both identifiers based on the same received data. Such a solution has, though, a major disadvantage. Imagine a scenario in which the application of the first endpoint generates an application message that also will be used as a shared state. A copy of this message is stored in the first endpoint. The message is then compressed and transmitted to the second endpoint where it will be decompressed. The second endpoint calculates the two identifiers and compares them. If the content of the message has (unintentionally) been modified or changed (*e.g.*, in the compression of the message in the first endpoint or during the transmission of the message), however, the two endpoints will have access to somewhat different shared states. Since the two identifiers are determined by the second endpoint based on the received data, the identifiers will still match even though the uncompressed message (shared state) that the second endpoint has access to differs from the shared state copy stored in the first endpoint. When the second endpoint then subsequently compresses a message intended to the first endpoint using its version of the uncompressed message (shared state), it is very likely that the decompression of the same message will fail or result in an erroneous decompressed message in the first endpoint since the *shared* state copy stored in the first endpoint differs from the version used in the compression by the second endpoint. This is in clear contrast to the Applicants' claimed invention, wherein **states** are **applicable to multiple** (*i.e.* at least two) messages communicated between the endpoints and includes endpoint-associated data. In addition, **both endpoints store** their respective copy of the **state**. More importantly, the first endpoint generates the first identifier based on the state in advance of any processing of the state (*e.g.*, compression or transmission) that could result in a modification of the contents of the state. As a consequence, the **first identifier** truly **reflects** the version of the **state stored** in the **first endpoint**. The second identifier, however, is calculated by the second endpoint based on the state version this endpoint has received. As a consequence, the **second identifier** truly **reflects** the version of the **state** received by the **second endpoint**. If the two identifiers match it implies that both endpoints have access to the same state data. The second endpoint now knows that it can use its version of the state for message processing. In addition, the first endpoint is informed of the successful reception of the correct state upon reception of the acknowledge identifier, which in turn has been transmitted by the second endpoint in response to a correspondence between the identifiers.

For the foregoing reasons, claim 27 is not anticipated by RFC 3321. Whereas independent claims 39 and 49 recite analogous limitations, they are also not anticipated by RFC 3321. Furthermore, whereas claims 28-38, 40-48 and 50-53 are dependent from claims 27, 39 and 49, respectively, and include the limitations thereof, they are also not anticipated by that reference.

* * *

CONCLUSION

In view of the foregoing remarks, the Applicants believe all of the claims currently pending in the Application to be in a condition for allowance. The Applicants, therefore, respectfully request that the Examiner withdraw all rejections and issue a Notice of Allowance for claims 27-53.

The Applicants request a telephonic interview if the Examiner has any questions or requires any additional information that would further or expedite the prosecution of the Application.

Respectfully submitted,

Roger Burleigh

Roger S. Burleigh
Registration No. 40,542

Date: October 8, 2009

Ericsson Inc.
6300 Legacy Drive, M/S EVR 1-C-11
Plano, Texas 75024

(972) 583-5799
roger.burleigh@ericsson.com